Introduction to Machine Learning

Introduction to Natural Language Processing

Andres Mendez-Vazquez

March 5, 2019

Outline

Outline

This is an Overlapping Field

We have different ones

- Computational linguistics in linguistics,
- Natural Language Processing (NLP) in computer science,
- Speech Recognition in electrical engineering,
- Computational psycholinguistics in psychology.

1940's and 1950's

Two Foundational Paradigms

- Finite State Automaton (FSA),
- Probabilistic Models

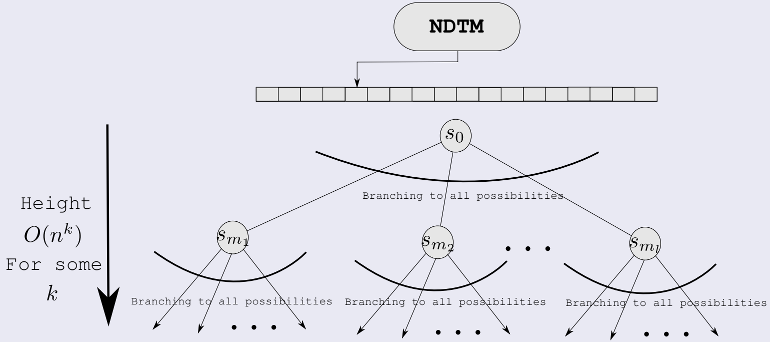
Remembering Turing

1940's and 1950's

Two Foundational Paradigms

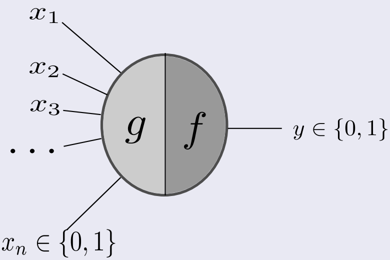
- Finite State Automaton (FSA),
- Probabilistic Models

Remembering Turing



Furthermore

McCulloch-Pitts Neuron



$$g\left(x_1, x_2, \ldots, x_n\right) = \bigoplus_i^n x_i$$

$$y = f\left(g\left(\boldsymbol{x}\right)\right) = \begin{cases} 1 & \text{if } g\left(\boldsymbol{x}\right) \geq \theta \\ 0 & \text{if } g\left(\boldsymbol{x}\right) < \theta \end{cases}$$

Formal Language Theory

Automata theory was contributed to by Shannon (1948)

- He applied probabilistic models of discrete Markov processes to automata for language.

Furthermore, Chomsky (1956)

- He used the finite-state Markov process from Shannon's work, to develop:
  - Finite-State Machines as a way to characterize a Grammar

$$\text{Context-Free Gramars} \cong \text{Finite-State Machine}$$

Formal Language Theory

Automata theory was contributed to by Shannon (1948)

- He applied probabilistic models of discrete Markov processes to automata for language.

Furthermore, Chomsky (1956)

- He used the finite-state Markov process from Shannon's work, to develop:
  - Finite-State Machines as a way to characterize a Grammar

$$\text{Context-Free Gramars} \cong \text{Finite-State Machine}$$

The Basis

For the following

- Context-Free Grammars
- Backus-Naur description

Basically

- Cobol
- Algol
- C
- etc

The Basis

For the following

- Context-Free Grammars
- Backus-Naur description

Basically

- Cobol
- Algol
- C
- etc

From the Side of Probability

From Shannon idea of noisy channels

- Probabilistic algorithms for speech and language processing.

Sound Spectrograph was Developed

- This led to the first machine speech recognizers in the early 1950's.

Bell Labs

- It Build a statistical system that could recognize any of the 10 digits from a single speaker

From the Side of Probability

From Shannon idea of noisy channels

- Probabilistic algorithms for speech and language processing.

Sound Spectrograph was Developed

- This led to the first machine speech recognizers in the early 1950's.

Bell Labs

- It Build a statistical system that could recognize any of the 10 digits from a single speaker

From the Side of Probability

From Shannon idea of noisy channels

- Probabilistic algorithms for speech and language processing.

Sound Spectrograph was Developed

- This led to the first machine speech recognizers in the early 1950's.

Bell Labs

- It Build a statistical system that could recognize any of the 10 digits from a single speaker

1957-1970

We had the Formal Language Processing Path

- The symbolic paradigm took off from two lines of research.

One leaded to...

- On parsing algorithms, initially top-down and bottom-up, and then via dynamic programming.

The Second one

- It leaded to the symbolic systems...

1957-1970

We had the Formal Language Processing Path

- The symbolic paradigm took off from two lines of research.

One leaded to...

- On parsing algorithms, initially top-down and bottom-up, and then via dynamic programming.

The Second one

- It leaded to the symbolic systems...

1957-1970

We had the Formal Language Processing Path

- The symbolic paradigm took off from two lines of research.

One leaded to...

- On parsing algorithms, initially top-down and bottom-up, and then via dynamic programming.

The Second one

- It leaded to the symbolic systems...

Furthermore

Something Notable

- The stochastic paradigm took hold mainly in departments of statistics and of electrical engineering.

For example

- Bayesian method was beginning to be applied to to the problem of optical character recognition.

A Famous Case

- Mosteller and Wallace (964) applied Bayesian methods to the problem of authorship attribution on The Federalist papers.

Furthermore

Something Notable

- The stochastic paradigm took hold mainly in departments of statistics and of electrical engineering.

For example

- Bayesian method was beginning to be applied to to the problem of optical character recognition.

A Famous Case

- Mosteller and Wallace (964) applied Bayesian methods to the problem of authorship attribution on The Federalist papers.

Furthermore

Something Notable

- The stochastic paradigm took hold mainly in departments of statistics and of electrical engineering.

For example

- Bayesian method was beginning to be applied to to the problem of optical character recognition.

A Famous Case

- Mosteller and Wallace (964) applied Bayesian methods to the problem of authorship attribution on The Federalist papers.

1970-1999

The Stochastic Paradigm

- It played a huge role in the development of speech recognition algorithms.

The Logic-Based Paradigm

Development of languages ad Prolog and Functional Grammars...

Natural Language Understanding

- It took off during this period

1970-1999

The Stochastic Paradigm

- It played a huge role in the development of speech recognition algorithms.

The Logic-Based Paradigm

Development of languages ad Prolog and Functional Grammars...

Natural Language Understanding

- It took off during this period

1970-1999

The Stochastic Paradigm

- It played a huge role in the development of speech recognition algorithms.

The Logic-Based Paradigm

Development of languages ad Prolog and Functional Grammars...

Natural Language Understanding

- It took off during this period

2000 - Present

We have many ways of representing documents

- Word2Vec, Singular Value Decomposition, Glove

New Retrieval Information Systems

- Better Search Methods for commercial Engines.

Sentiment Analysis

- It goes from a possibility to a reality

2000 - Present

We have many ways of representing documents

- Word2Vec, Singular Value Decomposition, Glove

New Retrieval Information Systems

- Better Search Methods for commercial Engines.

Sentiment Analysis

- It goes from a possibility to a reality

2000 - Present

We have many ways of representing documents

- Word2Vec, Singular Value Decomposition, Glove

New Retrieval Information Systems

- Better Search Methods for commercial Engines.

Sentiment Analysis

- It goes from a possibility to a reality

Outline

Document Representation

Imagine the following...

You have a bunch of documents... They are hundred thousands of them...

Then, we have an Opportunity or a Terrible Problem

How do you represent them in a easy way to handle them?

After all we want to

- Search them
- Compare them
- Rank them

Then, we have an Opportunity or a Terrible Problem

How do you represent them in a easy way to handle them?

After all we want to

- Search them
- Compare them
- Rank them

Then, we have an Opportunity or a Terrible Problem

How do you represent them in a easy way to handle them?

After all we want to

- Search them
- Compare them
- Rank them

Then, we have an Opportunity or a Terrible Problem

How do you represent them in a easy way to handle them?

After all we want to

- Search them
- Compare them
- Rank them

Question

How do we represent the meaning of a word?

- The idea that is represented by a word, phrase, etc.

Commonest linguistic way of thinking of meaning

- signifier (symbol) $\Leftrightarrow$ signified (idea or thing)
  - Denotational Semantics

Question

How do we represent the meaning of a word?

- The idea that is represented by a word, phrase, etc.

Commonest linguistic way of thinking of meaning

- signifier (symbol) $\Leftrightarrow$ signified (idea or thing)
  - Denotational Semantics

Then, How do we have a usable meaning?

Common Solution

- Use, for example, WordNet, a thesaurus containing lists of synonym sets and hypernyms

For Example

1. from nltk.corpus import wordnet as wn
2. poses = { 'n':'noun', 'v':'verb', 's':'adj (s)', 'a':'adj', 'r':'adv'}
3. for synset in wn.synsets("good"):
   print("{}: {}".format(poses[synset.pos()], ", ".join([l.name()\
   for l in synset.lemmas()])))

Then, How do we have a usable meaning?

Common Solution

- Use, for example, WordNet, a thesaurus containing lists of synonym sets and hypernyms

For Example

1. from nltk.corpus import wordnet as wn
2. poses = { 'n':'noun', 'v':'verb', 's':'adj (s)', 'a':'adj', 'r':'adv'}
3. for synset in wn.synsets("good"):

```
        print("{}: {}".format(poses[synset.pos()], ", ".join([l.name()\
        for l in synset.lemmas()])))
```

Problems with resources like WordNet

Great as a resource but missing stuff

- For Example, “proficient” is listed as a synonym for “good”. This is only correct in some contexts.

Missing new meanings of words

- wicked, badass, nifty, wizard, genius, ninja, bombest
- Impossible to keep up-to-date!

Furthermore

- Requires human labor to create and adapt
- It cannot compute accurate word similarity!!!

Problems with resources like WordNet

Great as a resource but missing stuff

- For Example, “proficient” is listed as a synonym for “good”. This is only correct in some contexts.

Missing new meanings of words

- wicked, badass, nifty, wizard, genius, ninja, bombest
- Impossible to keep up-to-date!

Furthermore

- Requires human labor to create and adapt
- It cannot compute accurate word similarity!!!

Problems with resources like WordNet

Great as a resource but missing stuff

- For Example, "proficient" is listed as a synonym for "good". This is only correct in some contexts.

Missing new meanings of words

- wicked, badass, nifty, wizard, genius, ninja, bombest
- Impossible to keep up-to-date!

Furthermore

- Requires human labor to create and adapt
- It cannot compute accurate word similarity!!!

Outline

Representing Words

In traditional NLP

- We regard words as discrete symbols.
  - hotel, conference, motel – a localist representation

Words can be represented by one-hot vectors

$$motel = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$hotel = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Dimension of those one-hot vectors

- Vector dimension = number of words in vocabulary

Representing Words

In traditional NLP

- We regard words as discrete symbols.
  - hotel, conference, motel – a localist representation

Words can be represented by one-hot vectors

$$motel = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$
$$hotel = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Dimension of those one-hot vectors

- Vector dimension = number of words in vocabulary

Representing Words

In traditional NLP

- We regard words as discrete symbols.
  - hotel, conference, motel – a localist representation

Words can be represented by one-hot vectors

$$motel = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$
$$hotel = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Dimension of those one-hot vectors

- Vector dimension = number of words in vocabulary

Problem with words as discrete symbols

In web search, if user searches for "Seattle motel"

- We would like to match documents containing "Seattle hotel"

However

$$motel = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$hotel = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

These two vectors are orthogonal

- There is no natural notion of similarity for one-hot vectors!

Problem with words as discrete symbols

In web search, if user searches for "Seattle motel"

- We would like to match documents containing "Seattle hotel"

However

$$
\begin{aligned}
motel &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \\
hotel &= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}
\end{aligned}
$$

These two vectors are orthogonal

- There is no natural notion of similarity for one-hot vectors!

Problem with words as discrete symbols

In web search, if user searches for "Seattle motel"

- We would like to match documents containing "Seattle hotel"

However

$$motel = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$
$$hotel = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

These two vectors are orthogonal

- There is no natural notion of similarity for one-hot vectors!

Solution

Question

- Could try to rely on WordNet's list of synonyms to get similarity?

But it is well-known to fail badly

- incompleteness, etc.

Instead

- learn to encode similarity in the vectors themselves

Solution

Question

- Could try to rely on WordNet's list of synonyms to get similarity?

But it is well-known to fail badly

- incompleteness, etc.

Instead

- learn to encode similarity in the vectors themselves

Solution

Question

- Could try to rely on WordNet's list of synonyms to get similarity?

But it is well-known to fail badly

- incompleteness, etc.

Instead

- learn to encode similarity in the vectors themselves

Outline

Empty

A word's meaning is given by the words that frequently appear close-by

- "You shall know a word by the company it keeps" (J. R. Firth 1957: 11)
- One of the most successful ideas of modern statistical NLP!

When a word w appears in a text

- its context is the set of words that appear nearby (within a fixed-size window.

Empty

A word's meaning is given by the words that frequently appear close-by

- "You shall know a word by the company it keeps" (J. R. Firth 1957: 11)
- One of the most successful ideas of modern statistical NLP!

When a word w appears in a text

- its context is the set of words that appear nearby (within a fixed-size window.

Furthermore

Use the many contexts of w to build up a representation of w

- ...government debt problems turning into **banking** crises as happened in 2009...
- ...saying that Europe needs unified **banking** regulation to replace the hodgepodge...
- ...India has just given its **banking** system a shot in the arm...

What do we want?

We want to build a dense vector for each word

- so similar words appear in similar contexts

$$banking = \begin{pmatrix} 0.26 \\ 0.72 \\ -0.177 \\ -0.107 \\ 0.018 \\ 0.271 \end{pmatrix}$$

Word Vectors are sometimes called word embeddings or word representations

- They are a distributed representation

What do we want?

We want to build a dense vector for each word

- so similar words appear in similar contexts

$$banking = \begin{pmatrix} 0.26 \\ 0.72 \\ -0.177 \\ -0.107 \\ 0.018 \\ 0.271 \end{pmatrix}$$

Word Vectors are sometimes called word embeddings or word representations

- They are a distributed representation

Outline

Sparse Array

First

We will start with sparse one-dimensional arrays, which are simpler

Example

	0	1	2	3	4	5	6	7	8	9	10	11
Array	0	0	0	0	17	0	0	23	14	0	0	0

Sparse Array

First

We will start with sparse one-dimensional arrays, which are simpler

Example

	0	1	2	3	4	5	6	7	8	9	10	11
Array	0	0	0	0	**17**	0	0	**23**	**14**	0	0	0

Representation

We can use the following representation

$$\rightarrow \boxed{4 \mid 17 \mid \;\;} \rightarrow \boxed{7 \mid 23 \mid \;\;} \rightarrow \boxed{8 \mid 14}$$

Representation

We can use the following representation

$$\rightarrow \boxed{4 \mid 17 \mid \quad} \rightarrow \boxed{7 \mid 23 \mid \quad} \rightarrow \boxed{8 \mid 14}$$

Where

1. The front element is the index.
2. The second element is the value at cell index.
3. A pointer to the next element

Representation

We can use the following representation

$$\rightarrow \boxed{4 \mid 17 \mid \quad} \rightarrow \boxed{7 \mid 23 \mid \quad} \rightarrow \boxed{8 \mid 14}$$

Where

1. The front element is the index.
2. The second element is the value at cell index.
3. A pointer to the next element

Representation

We can use the following representation

$\rightarrow \boxed{4 \mid 17 \mid \;} \rightarrow \boxed{7 \mid 23 \mid \;} \rightarrow \boxed{8 \mid 14}$

Where

1. The front element is the index.
2. The second element is the value at cell index.
3. A pointer to the next element

However what is the Complexity?

To find an element different from Zero

- We need to iterate through the list!!!

Therefore, we have

- $O(m)$ with $m =$ the number of elements different of zero.

We need something different

- What?

However what is the Complexity?

To find an element different from Zero

- We need to iterate through the list!!!

Therefore, we have

- $O(m)$ with $m =$ the number of elements different of zero.

We need something different

- What?

However what is the Complexity?

To find an element different from Zero

- We need to iterate through the list!!!

Therefore, we have

- $O(m)$ with $m =$ the number of elements different of zero.

We need something different

- What?

Given

A sparse vector

$$\boldsymbol{x}^T = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix}$$

Then, for each

$$\begin{array}{cc} value & \\ index & \end{array}\begin{bmatrix} x_{i_1} & x_{i_2} & x_{i_3} & x_{i_4} & x_{i_5} \\ i_1 & i_2 & i_3 & i_4 & i_5 \end{bmatrix}$$

Therefore, we can use

- Binary search on the indexes to find elements in the structure!!!

Given

A sparse vector

$$\boldsymbol{x}^T = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix}$$

Then, for each

$$\begin{array}{c} value \\ index \end{array} \begin{bmatrix} x_{i_1} & x_{i_2} & x_{i_3} & x_{i_4} & x_{i_5} \\ i_1 & i_2 & i_3 & i_4 & i_5 \end{bmatrix}$$

Therefore, we can use

- Binary search on the indexes to find elements in the structure!!!

Given

A sparse vector

$$\boldsymbol{x}^T = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix}$$

Then, for each

$$\begin{array}{c} value \\ index \end{array} \begin{bmatrix} x_{i_1} & x_{i_2} & x_{i_3} & x_{i_4} & x_{i_5} \\ i_1 & i_2 & i_3 & i_4 & i_5 \end{bmatrix}$$

Therefore, we can use

- Binary search on the indexes to find elements in the structure!!!

Compressed Sparse Row

Format Compressed Sparse Row (CSR)

AA (Values)	2	1	5	3	4	6	7	8	9	10	11	12	
JA(Column Indeces)	4	1	4	1	2	1	3	4	5	3	4	5	
	↑		↑			↑				↑		↑	
IA(Pointer Row i)	1		3			6				10		12	13

Example of usage

- Computing $c = Ab$

1. $c = 0$
2. for $i = 1$ to n
3. for $j = IA(i)$ to $IA(i+1) - 1$
4. $c_i = c_i + AA(j) \times b_{JA(j)}$

Compressed Sparse Row

Format Compressed Sparse Row (CSR)

AA (Values)	2	1	5	3	4	6	7	8	9	10	11	12	
JA(Column Indeces)	4	1	4	1	2	1	3	4	5	3	4	5	
	$\uparrow$		$\uparrow$			$\uparrow$				$\uparrow$		$\uparrow$	
IA(Pointer Row i)	1		3			6				10		12	13

Example of usage

- Computing $\boldsymbol{c} = A\boldsymbol{b}$

1. $\boldsymbol{c} = 0$
2. for $i = 1$ to n
3. for $i = IA\left(i\right)$ to $IA\left(i+1\right) - 1$
4. $\boldsymbol{c}_i = \boldsymbol{c}_i + AA\left(j\right) \times b_{JA(j)}$

There are many other ways

To compress the sparse matrices

- However, they are out of the scope of this class.

Outline

Why do we look at this?

Because we use NumPy to have better array operations

NumPy is the fundamental package for scientific computing with Python

- Sophisticated (broadcasting) functions tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Why do we look at this?

Because we use NumPy to have better array operations

NumPy is the fundamental package for scientific computing with Python

It contains among other things

- A powerful N-dimensional array object.
- Sophisticated (broadcasting) functions tools for integrating C/C++ and Fortran code.
- Useful linear algebra, Fourier transform, and random number capabilities

Why do we look at this?

Because we use NumPy to have better array operations

NumPy is the fundamental package for scientific computing with Python

It contains among other things

- A powerful N-dimensional array object.
- Sophisticated (broadcasting) functions tools for integrating C/C++ and Fortran code.
- Useful linear algebra, Fourier transform, and random number capabilities

Why do we look at this?

Because we use NumPy to have better array operations

NumPy is the fundamental package for scientific computing with Python

It contains among other things

- A powerful N-dimensional array object.
- Sophisticated (broadcasting) functions tools for integrating C/C++ and Fortran code.
- Useful linear algebra, Fourier transform, and random number capabilities

But Why?

NumPy targets the CPython reference implementation of Python

- Mathematical algorithms written for this version of Python often run much slower than compiled equivalents.

  - Multidimensional arrays
  - Functions
  - Operators

Operating on those arrays

But Why?

NumPy targets the CPython reference implementation of Python

- Mathematical algorithms written for this version of Python often run much slower than compiled equivalents.

Therefore

- NumPy addresses the slowness problem partly by providing:
  - Multidimensional arrays
  - Functions
  - Operators

  Operating on those arrays.

But Why?

NumPy targets the CPython reference implementation of Python

- Mathematical algorithms written for this version of Python often run much slower than compiled equivalents.

Therefore

- NumPy addresses the slowness problem partly by providing:
  - Multidimensional arrays
  - Functions
  - Operators

  Operating on those arrays.

But Why?

NumPy targets the CPython reference implementation of Python

- Mathematical algorithms written for this version of Python often run much slower than compiled equivalents.

Therefore

- NumPy addresses the slowness problem partly by providing:
  - Multidimensional arrays
  - Functions
  - Operators

  Operating on those arrays.

But Why?

NumPy targets the CPython reference implementation of Python

- Mathematical algorithms written for this version of Python often run much slower than compiled equivalents.

Therefore

- NumPy addresses the slowness problem partly by providing:
  - Multidimensional arrays
  - Functions
  - Operators

  Operating on those arrays.

But Why?

NumPy targets the CPython reference implementation of Python

- Mathematical algorithms written for this version of Python often run much slower than compiled equivalents.

Therefore

- NumPy addresses the slowness problem partly by providing:
  - Multidimensional arrays
  - Functions
  - Operators

  Operating on those arrays.

The ndarray data structure

The core functionality of NumPy is its "ndarray"

- They are n-dimensional array, data structure.

These arrays are strided views on memory

- Basically describing the gaps on memory for the array structure!!!

What are you saying?

- Let me show this... at the blackboard...

The ndarray data structure

The core functionality of NumPy is its "ndarray"

- They are n-dimensional array, data structure.

These arrays are strided views on memory

- Basically describing the gaps on memory for the array structure!!!

What are you saying?

- Let me show this... at the blackboard...

The ndarray data structure

The core functionality of NumPy is its "ndarray"

- They are n-dimensional array, data structure.

These arrays are strided views on memory

- Basically describing the gaps on memory for the array structure!!!

What are you saying?

- Let me show this... at the blackboard...

Outline

However...

Numerical Recipes: The Art of Scientific Computing, Third Edition, 2007

- “It is wasteful to use general methods of linear algebra on such problems, because most of the $O\left(N^3\right)$ arithmetic operations devoted to solving the set of equations or inverting the matrix involve zero operands.”

- Compressed Sparse Row (CSR): The sparse matrix is represented using three one-dimensional arrays for the non-zero values, the extents of the rows, and the column indexes.
- Compressed Sparse Column (CSC): The same as the Compressed Sparse Row method except the column indices are compressed and read first before the row indices.

However...

Numerical Recipes: The Art of Scientific Computing, Third Edition, 2007

- "It is wasteful to use general methods of linear algebra on such problems, because most of the $O\left(N^3\right)$ arithmetic operations devoted to solving the set of equations or inverting the matrix involve zero operands."

Therefore

- There are multiple data structures that can be used to efficiently construct a sparse matrix:
  - **Compressed Sparse Row** (CSR). The sparse matrix is represented using three one-dimensional arrays for the non-zero values, the extents of the rows, and the column indexes.
  - **Compressed Sparse Column** (CSC). The same as the Compressed Sparse Row method except the column indices are compressed and read first before the row indices.

However...

Numerical Recipes: The Art of Scientific Computing, Third Edition, 2007

- "It is wasteful to use general methods of linear algebra on such problems, because most of the $O\left(N^3\right)$ arithmetic operations devoted to solving the set of equations or inverting the matrix involve zero operands."

Therefore

- There are multiple data structures that can be used to efficiently construct a sparse matrix:
  - **Compressed Sparse Row** (CSR). The sparse matrix is represented using three one-dimensional arrays for the non-zero values, the extents of the rows, and the column indexes.
  - **Compressed Sparse Column** (CSC). The same as the Compressed Sparse Row method except the column indices are compressed and read first before the row indices.

However...

Numerical Recipes: The Art of Scientific Computing, Third Edition, 2007

- "It is wasteful to use general methods of linear algebra on such problems, because most of the $O\left(N^3\right)$ arithmetic operations devoted to solving the set of equations or inverting the matrix involve zero operands."

Therefore

- There are multiple data structures that can be used to efficiently construct a sparse matrix:
    - **Compressed Sparse Row** (CSR). The sparse matrix is represented using three one-dimensional arrays for the non-zero values, the extents of the rows, and the column indexes.
    - **Compressed Sparse Column** (CSC). The same as the Compressed Sparse Row method except the column indices are compressed and read first before the row indices.

For Example

In Python, we have

- class scipy.sparse.coo_matrix

This is a coordinate format

- Also known as the 'ijv' or 'triplet' format.

Where

- i = the row index
- j = the column index
- v = value

For Example

In Python, we have

- class scipy.sparse.coo_matrix

This is a coordinate format

- Also known as the 'ijv' or 'triplet' format.

Where

- i = the row index
- j = the column index
- v = value

For Example

In Python, we have

- class scipy.sparse.coo_matrix

This is a coordinate format

- Also known as the 'ijv' or 'triplet' format.

Where

- i = the row index
- j = the column index
- v = value

This can be instantiated in several way

Using Dense Matrix D

- coo_matrix(D) with a dense matrix D

Using a Sparse Matrix S

- coo_matrix(S) with another sparse matrix S (equivalent to S.tocoo())

Also, we have

- coo_matrix((M, N), [dtype]) to construct an empty matrix with shape (M, N)

This can be instantiated in several way

Using Dense Matrix D

- coo_matrix(D) with a dense matrix D

Using a Sparse Matrix S

- coo_matrix(S) with another sparse matrix S (equivalent to S.tocoo())

Also, we have

- coo_matrix((M, N), [dtype]) to construct an empty matrix with shape (M, N)

This can be instantiated in several way

Using Dense Matrix D

- coo_matrix(D) with a dense matrix D

Using a Sparse Matrix S

- coo_matrix(S) with another sparse matrix S (equivalent to S.tocoo())

Also, we have

- coo_matrix((M, N), [dtype]) to construct an empty matrix with shape (M, N)

And one that we like

coo_matrix((data, (i, j)), [shape=(M, N)]) to construct from three arrays

- data[:] the entries of the matrix,
- in any order i[:] the row indices of the matrix entries
- j[:] the column indices of the matrix entries

And one that we like

coo_matrix((data, (i, j)), [shape=(M, N)]) to construct from three arrays

- data[:] the entries of the matrix,
- in any order i[:] the row indices of the matrix entries
- j[:] the column indices of the matrix entries

And one that we like

coo_matrix((data, (i, j)), [shape=(M, N)]) to construct from three arrays

- data[:] the entries of the matrix,
- in any order i[:] the row indices of the matrix entries
- j[:] the column indices of the matrix entries

Then

Conversion for Indexing

1. COO is a fast format for constructing sparse matrices
2. Once a matrix has been constructed,
    1. convert to CSR or CSC format for fast arithmetic and matrix vector operations

Example

Example

$$\begin{bmatrix} 0 & 0 & 3 & 0 & 4 \\ 0 & 0 & 5 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 6 & 0 & 0 \end{bmatrix}$$

Thus

list =

row	1	1	2	2	4	4
column	3	5	3	4	2	3
value	3	4	5	6	2	6

Example

Example

$$\begin{bmatrix} 0 & 0 & 3 & 0 & 4 \\ 0 & 0 & 5 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 6 & 0 & 0 \end{bmatrix}$$

Thus

$$\text{list} = \begin{array}{c|cccccc|} \text{row} & 1 & 1 & 2 & 2 & 4 & 4 \\ \text{column} & 3 & 5 & 3 & 4 & 2 & 3 \\ \text{value} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{6} & \mathbf{2} & \mathbf{6} \end{array}$$

Outline

Latent Semantic Indexing (LSI)

What is it?

- It is a method for discovering hidden concepts in document data

Representation

- Each document and term (word) is then expressed as a vector with elements corresponding to these concepts.

Thus

- Each element in a vector gives the degree of participation of the document or term in the corresponding concept.

Latent Semantic Indexing (LSI)

What is it?

- It is a method for discovering hidden concepts in document data

Representation

- Each document and term (word) is then expressed as a vector with elements corresponding to these concepts.

Thus

- Each element in a vector gives the degree of participation of the document or term in the corresponding concept.

Latent Semantic Indexing (LSI)

What is it?

- It is a method for discovering hidden concepts in document data

Representation

- Each document and term (word) is then expressed as a vector with elements corresponding to these concepts.

Thus

- Each element in a vector gives the degree of participation of the document or term in the corresponding concept.

Goal

We want numerical representations

- The goal is not to describe the concepts verbally

We want

- to be able to represent the documents and terms in a unified way

For exposing

- Document-Dwocument, Document-Term, and Term-Term similarities or semantic relationship which are otherwise hidden.

Goal

We want numerical representations

- The goal is not to describe the concepts verbally

We want

- to be able to represent the documents and terms in a unified way

For exposing

- Document-Dwocument, Document-Term, and Term-Term similarities or semantic relationship which are otherwise hidden.

Goal

We want numerical representations

- The goal is not to describe the concepts verbally

We want

- to be able to represent the documents and terms in a unified way

For exposing

- Document-Dwocument, Document-Term, and Term-Term similarities or semantic relationship which are otherwise hidden.

Outline

Example

Suppose we have the following set of five documents

- d_1 : Romeo and Juliet.
- d_2 : Juliet: O happy dagger!
- d_3 : Romeo died by dagger.
- d_4: "Live free or die", that's the New-Hampshire's motto.
- d_5 : Did you know, New-Hampshire is in New-England.

And you have a search query: "dies, dagger"

- Clearly, document d_3 should be ranked top of the query given it contains both dies and dager.

Then

- d_2 and d_4 should follow, each containing a word of the query.

Example

Suppose we have the following set of five documents

- d_1 : Romeo and Juliet.
- d_2 : Juliet: O happy dagger!
- d_3 : Romeo died by dagger.
- d_4: "Live free or die", that's the New-Hampshire's motto.
- d_5 : Did you know, New-Hampshire is in New-England.

And you have a search query: "dies, dagger"

- Clearly, document d_3 should be ranked top of the query given it contains both dies and dager.

Then

- d_2 and d_4 should follow, each containing a word of the query.

Example

Suppose we have the following set of five documents

- d_1 : Romeo and Juliet.
- d_2 : Juliet: O happy dagger!
- d_3 : Romeo died by dagger.
- d_4: "Live free or die", that's the New-Hampshire's motto.
- d_5 : Did you know, New-Hampshire is in New-England.

And you have a search query: "dies, dagger"

- Clearly, document d_3 should be ranked top of the query given it contains both dies and dager.

Then

- d_2 and d_4 should follow, each containing a word of the query.

Nevertheless

However, what about d_1 and d_5?

- As humans we know that d_1 is quite related to the query.
- On the other hand, d_5 is not so much related to the query.

Thus, we would like d_1 but not d_5

- We want d_1 to be ranked higher than d_5.

Can the machine deduce this?

- The answer is yes, LSI does exactly that.

Nevertheless

However, what about d_1 and d_5?

- As humans we know that d_1 is quite related to the query.
- On the other hand, d_5 is not so much related to the query.

Thus, we would like d_1 but not d_5

- We want d_1 to be ranked higher than d_5.

Can the machine deduce this?

- The answer is yes, LSI does exactly that.

Nevertheless

However, what about d_1 and d_5?

- As humans we know that d_1 is quite related to the query.
- On the other hand, d_5 is not so much related to the query.

Thus, we would like d_1 but not d_5

- We want d_1 to be ranked higher than d_5.

Can the machine deduce this?

- The answer is yes, LSI does exactly that.

Occurrence Matrix

The occurrence matrix A be the $m \times n$ term-document matrix of a collection of documents

- Each column of A corresponds to a document.

We fill such matrix in the following way

- If term i occurs a times in document j then $A[i, j] = a$.

The dimensions of A, m and n,

- They correspond to the number of words and documents, respectively, in the collection.

Occurrence Matrix

The occurrence matrix A be the $m \times n$ term-document matrix of a collection of documents

- Each column of A corresponds to a document.

We fill such matrix in the following way

- If term i occurs a times in document j then $A[i,j] = a$.

The dimensions of A, m and n,

- They correspond to the number of words and documents, respectively, in the collection.

Occurrence Matrix

The occurrence matrix A be the $m \times n$ term-document matrix of a collection of documents

- Each column of A corresponds to a document.

We fill such matrix in the following way

- If term i occurs a times in document j then $A[i, j] = a$.

The dimensions of A, m and n,

- They correspond to the number of words and documents, respectively, in the collection.

For Example

We have the following matrix

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
ne-hamshire	0	0	0	1	1

Outline

Also Known as Karhunen-Loeve Transform

Setup

- Consider a data set of observations $\{\boldsymbol{x}_n\}$ with $n = 1, 2, ..., N$ and $\boldsymbol{x_n} \in R^d$.

Goal

Project data onto space with dimensionality $m < d$ (We assume m is given)

Also Known as Karhunen-Loeve Transform

Setup

- Consider a data set of observations $\{\boldsymbol{x}_n\}$ with $n = 1, 2, ..., N$ and $\boldsymbol{x_n} \in R^d$.

Goal

Project data onto space with dimensionality $m < d$ (We assume m is given)

Dimensional Variance

Remember the Sample Variance Sample

$$VAR(X) = \frac{\sum_{i=1}^{N} (x_i - \overline{x})(x_i - \overline{x})}{N-1} \quad (1)$$

You can do the same in the case of two variables X and Y

$$COV(X,Y) = \frac{\sum_{i=1}^{N} (x_i - \overline{x})(y_i - \overline{y})}{N-1} \quad (2)$$

Dimensional Variance

Remember the Sample Variance Sample

$$VAR(X) = \frac{\sum_{i=1}^{N} (x_i - \overline{x})(x_i - \overline{x})}{N-1} \quad (1)$$

You can do the same in the case of two variables X and Y

$$COV(X, Y) = \frac{\sum_{i=1}^{N} (x_i - \overline{x})(y_i - \overline{y})}{N-1} \quad (2)$$

Now, Define

Given the data

$$\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N \tag{3}$$

where $\boldsymbol{x}_i$ is a column vector

Construct the sample mean

$$\overline{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \tag{4}$$

Build new matrix

$$\boldsymbol{x}_1 - \overline{\boldsymbol{x}}, \boldsymbol{x}_2 - \overline{\boldsymbol{x}}, ..., \boldsymbol{x}_N - \overline{\boldsymbol{x}} \tag{5}$$

Now, Define

Given the data

$$\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N \tag{3}$$

where $\boldsymbol{x}_i$ is a column vector

Construct the sample mean

$$\overline{\boldsymbol{x}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i \tag{4}$$

Build new matrix

$$\boldsymbol{x}_1 - \overline{\boldsymbol{x}}, \boldsymbol{x}_2 - \overline{\boldsymbol{x}}, ..., \boldsymbol{x}_N - \overline{\boldsymbol{x}} \tag{5}$$

Now, Define

Given the data

$$\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N \tag{3}$$

where $\boldsymbol{x}_i$ is a column vector

Construct the sample mean

$$\overline{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \tag{4}$$

Build new matrix

$$\boldsymbol{x}_1 - \overline{\boldsymbol{x}}, \boldsymbol{x}_2 - \overline{\boldsymbol{x}}, ..., \boldsymbol{x}_N - \overline{\boldsymbol{x}} \tag{5}$$

Build the New Data Matrix

We have the following data matrix

$$X = \begin{pmatrix} (\boldsymbol{x}_1 - \overline{\boldsymbol{x}})^T \\ (\boldsymbol{x}_2 - \overline{\boldsymbol{x}})^T \\ \vdots \\ (\boldsymbol{x}_N - \overline{\boldsymbol{x}})^T \end{pmatrix}$$

Build the Sample Covariance

The Sample Covariance Matrix

$$S = \frac{1}{N-1} X^T X \tag{6}$$

Properties

1. The ijth value of S is equivalent to σ_{ij}^2.
2. The iith value of S is equivalent to σ_{ii}^2.
3. What else? Look at a plane Center and Rotating!!!

Build the Sample Covariance

The Sample Covariance Matrix

$$S = \frac{1}{N-1} X^T X \quad (6)$$

Properties

1. The ijth value of S is equivalent to σ_{ij}^2.
2. The iith value of S is equivalent to σ_{ii}^2.
3. What else? Look at a plane Center and Rotating!!!

Outline

This Reapers in the Document-Document Space

Basically the document-document matrix is a covariance matrix

$$B_{dd} = A^T A$$

If documents i and j have b words in common then $B[i,j] = b$

$$B = A^T A = \begin{pmatrix} 2 & 1 & 1 & 0 & 0 \\ 1 & 3 & 1 & 0 & 0 \\ 1 & 1 & 3 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

This Reapers in the Document-Document Space

Basically the document-document matrix is a covariance matrix

$$B_{dd} = A^T A$$

If documents i and j have b words in common then $B[i,j] = b$

$$B = A^T A = \begin{pmatrix} 2 & 1 & 1 & 0 & 0 \\ 1 & 3 & 1 & 0 & 0 \\ 1 & 1 & 3 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Similarly, in the case of the Term-Term Space

On the other hand

$$C = AA^T$$

If terms i and j occur together in c documents then $C[i,j] = c$

$$C = AA^T = \begin{pmatrix} 2 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 2 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 \end{pmatrix}$$

Similarly, in the case of the Term-Term Space

On the other hand

$$C = AA^T$$

If terms i and j occur together in c documents then $C[i,j] = c$

$$C = AA^T = \begin{pmatrix} 2 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 2 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 \end{pmatrix}$$

Outline

Now, we have...

A be an $m \times n$ matrix with entries being real numbers and $m > n$

- It has been shown that the eigenvalues of such matrices $A^T A$ are real non-negative numbers.

$$\sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_n^2$$

For some index r (possibly n)

- The first r numbers $\sigma_1, \sigma_2, \ldots, \sigma_r$ are positive whereas the rest are zero.

We also know that

- The corresponding eigenvectors $x_1, x_2, \ldots, x_r$ are perpendicular, and we normalize to have length one.

Now, we have...

A be an $m \times n$ matrix with entries being real numbers and $m > n$

- It has been shown that the eigenvalues of such matrices $A^T A$ are real non-negative numbers.

$$\sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_n^2$$

For some index r (possibly n)

- The first r numbers $\sigma_1, \sigma_2, ..., \sigma_r$ are positive whereas the rest are zero.

We also know that

- The corresponding eigenvectors $x_1, x_2, ..., x_r$ are perpendicular, and we normalize to have length one.

Now, we have...

A be an $m \times n$ matrix with entries being real numbers and $m > n$

- It has been shown that the eigenvalues of such matrices $A^T A$ are real non-negative numbers.

$$\sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_n^2$$

For some index r (possibly n)

- The first r numbers $\sigma_1, \sigma_2, ..., \sigma_r$ are positive whereas the rest are zero.

We also know that

- The corresponding eigenvectors $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_r$ are perpendicular, and we normalize to have length one.

Thus

We have that

$$S_1 = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_r]$$

Now, we create the vectors

$$y_1 = \frac{1}{\sigma_1} A x_1, ..., y_r = \frac{1}{\sigma_r} A x_r$$

Thus

We have that

$$S_1 = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_r]$$

Now, we create the vectors

$$\boldsymbol{y}_1 = \frac{1}{\sigma_1} A\boldsymbol{x}_1, ..., \boldsymbol{y}_r = \frac{1}{\sigma_r} A\boldsymbol{x}_r$$

They are perpendicular to each other

Given that

$$\boldsymbol{y}_i^T \boldsymbol{y}_j = \left(\frac{1}{\sigma_i} A\boldsymbol{x}_i\right)^T \left(\frac{1}{\sigma_j} A\boldsymbol{x}_j\right)$$

We have then

$$\boldsymbol{y}_i^T \boldsymbol{y}_j = \frac{1}{\sigma_i \sigma_j} \boldsymbol{x}_i^T B \boldsymbol{x}_j = \frac{1}{\sigma_i \sigma_j} \boldsymbol{x}_i^T \sigma_j^2 \boldsymbol{x}_j = \frac{\sigma_j}{\sigma_i} \boldsymbol{x}_i^T \boldsymbol{x}_j$$

They are perpendicular to each other

Given that

$$\boldsymbol{y}_i^T \boldsymbol{y}_j = \left(\frac{1}{\sigma_i} A\boldsymbol{x}_i\right)^T \left(\frac{1}{\sigma_j} A\boldsymbol{x}_j\right)$$

We have then

$$\boldsymbol{y}_i^T \boldsymbol{y}_j = \frac{1}{\sigma_i \sigma_j} \boldsymbol{x}_i^T B \boldsymbol{x}_j = \frac{1}{\sigma_i \sigma_j} \boldsymbol{x}_i^T \sigma_j^2 \boldsymbol{x}_j = \frac{\sigma_j}{\sigma_i} \boldsymbol{x}_i^T \boldsymbol{x}_j$$

With the following values

We have the following

$$\boldsymbol{y}_i^T \boldsymbol{y}_j = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

In a similar way, we have

$$S_2 = [\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_r]$$

With the following values

We have the following

$$\boldsymbol{y}_i^T \boldsymbol{y}_j = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

In a similar way, we have

$$S_2 = [\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_r]$$

Therefore, we have the following...

We have

$$\boldsymbol{y}_j^T A \boldsymbol{x}_i = \boldsymbol{y}_j^T \left(\sigma_i \boldsymbol{y}_i\right) = \sigma_i \boldsymbol{y}_j^T \boldsymbol{y}_i$$

Then, we have

$$\boldsymbol{y}_j^T A \boldsymbol{x}_i = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

From this we have

$$S_2^T A S_1 = \Sigma$$

Therefore, we have the following...

We have

$$\boldsymbol{y}_j^T A \boldsymbol{x}_i = \boldsymbol{y}_j^T \left(\sigma_i \boldsymbol{y}_i\right) = \sigma_i \boldsymbol{y}_j^T \boldsymbol{y}_i$$

Then, we have

$$\boldsymbol{y}_j^T A \boldsymbol{x}_i = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

From this we have

$$S_2^T A S_1 = \Sigma$$

Therefore, we have the following...

We have

$$\boldsymbol{y}_j^T A \boldsymbol{x}_i = \boldsymbol{y}_j^T \left(\sigma_i \boldsymbol{y}_i\right) = \sigma_i \boldsymbol{y}_j^T \boldsymbol{y}_i$$

Then, we have

$$\boldsymbol{y}_j^T A \boldsymbol{x}_i = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

From this we have

$$S_2^T A S_1 = \Sigma$$

What is Σ?

We have the following matrix

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r \end{pmatrix}$$

This are called the singular values

- They are the square roots of the eigenvalues of $A^T A$ and totally determined by A

What is Σ?

We have the following matrix

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r \end{pmatrix}$$

This are called the singular values

- They are the square roots of the eigenvalues of $A^T A$ and totally determined by A

Using a little bit of notation

We have the following notation

$$A = S\Sigma U^T$$

Going Back to the Documents and Terms

- Clearly, both B and C are square and symmetric:
  - B is an $m \times m$ matrix
  - C is an $n \times n$ matrix

Thus, we perform the singular value decomposition

$$\Sigma = \begin{pmatrix} 2.285 & 0 & 0 & 0 & 0 \\ 0 & 2.010 & 0 & 0 & 0 \\ 0 & 0 & 1.361 & 0 & 0 \\ 0 & 0 & 0 & 1.118 & 0 \\ 0 & 0 & 0 & 0 & 0.797 \end{pmatrix}$$

Using a little bit of notation

We have the following notation

$$A = S\Sigma U^T$$

Going Back to the Documents and Terms

- Clearly, both B and C are square and symmetric:
  - B is an $m \times m$ matrix
  - C is an $n \times n$ matrix

Thus, we perform the singular value decomposition

$$\Sigma = \begin{pmatrix} 2.285 & 0 & 0 & 0 & 0 \\ 0 & 2.010 & 0 & 0 & 0 \\ 0 & 0 & 1.361 & 0 & 0 \\ 0 & 0 & 0 & 1.118 & 0 \\ 0 & 0 & 0 & 0 & 0.797 \end{pmatrix}$$

Using a little bit of notation

We have the following notation

$$A = S\Sigma U^T$$

Going Back to the Documents and Terms

- Clearly, both B and C are square and symmetric:
  - B is an $m \times m$ matrix
  - C is an $n \times n$ matrix

Thus, we perform the singular value decomposition

$$\Sigma = \begin{pmatrix} 2.285 & 0 & 0 & 0 & 0 \\ 0 & 2.010 & 0 & 0 & 0 \\ 0 & 0 & 1.361 & 0 & 0 \\ 0 & 0 & 0 & 1.118 & 0 \\ 0 & 0 & 0 & 0 & 0.797 \end{pmatrix}$$

What about singular values "too small"

What really constitutes "too small"

- It is usually determined empirically.
  - For example for large documents "300"

Therefore, you keep k of them

$$A_k = S_k \Sigma_k U_k^T$$

Observe that since S_k, Σ_k, U_k^T

$$m \times k \cdot k \times k \cdot k \times n = m \times n$$

What about singular values "too small"

What really constitutes "too small"

- It is usually determined empirically.
  - For example for large documents "300"

Therefore, you keep k of them

$$A_k = S_k \Sigma_k U_k^T$$

Observe that since S_k, Σ_k, U_k^T

$$m \times k \cdot k \times k \cdot k \times n = m \times n$$

What about singular values "too small"

What really constitutes "too small"

- It is usually determined empirically.
  - For example for large documents "300"

Therefore, you keep k of them

$$A_k = S_k \Sigma_k U_k^T$$

Observe that since S_k, Σ_k, U_k^T

$$m \times k \cdot k \times k \cdot k \times n = m \times n$$

Intuitively, the k remaining ingredients of the eigenvectors in S and U

They correspond to the k "hidden concepts"

- where the terms and documents participate.

The terms are represented by the row vectors of the $m \times k$ matrix

$$S_k \Sigma_k$$

The documents by the column vectors the $k \times n$ matrix

$$\Sigma_k U_k^T$$

Intuitively, the k remaining ingredients of the eigenvectors in S and U

They correspond to the k "hidden concepts"

- where the terms and documents participate.

The terms are represented by the row vectors of the $m \times k$ matrix

$$S_k \Sigma_k$$

The documents by the column vectors the $k \times n$ matrix

$$\Sigma_k U_k^T$$

Intuitively, the k remaining ingredients of the eigenvectors in S and U

They correspond to the k "hidden concepts"

- where the terms and documents participate.

The terms are represented by the row vectors of the $m \times k$ matrix

$$S_k \Sigma_k$$

The documents by the column vectors the $k \times n$ matrix

$$\Sigma_k U_k^T$$

Then

The query is represented by the centroid of the vectors for its terms

- Basically, we find the representative vectors of the query and use the centroids

How?

$$c = \frac{1}{t}\sum_{i=1}^{t} q_i$$

Then, we have the cosine distance

$$s(x, y) = \cos(\theta) = \frac{x^T y}{\|x\| \, \|y\|}$$

Then

The query is represented by the centroid of the vectors for its terms

- Basically, we find the representative vectors of the query and use the centroids

How?

$$\boldsymbol{c} = \frac{1}{t}\sum_{i=1}^{t} \boldsymbol{q}_i$$

Then, we have the cosine distance

$$s\left(\boldsymbol{x}, \boldsymbol{y}\right) = \cos\left(\theta\right) = \frac{\boldsymbol{x}^T \boldsymbol{y}}{\left\|\boldsymbol{x}\right\| \left\|\boldsymbol{y}\right\|}$$

Then

The query is represented by the centroid of the vectors for its terms

- Basically, we find the representative vectors of the query and use the centroids

How?

$$\boldsymbol{c} = \frac{1}{t}\sum_{i=1}^{t} \boldsymbol{q}_i$$

Then, we have the cosine distance

$$s\left(\boldsymbol{x}, \boldsymbol{y}\right) = \cos\left(\theta\right) = \frac{\boldsymbol{x}^T \boldsymbol{y}}{\left\|\boldsymbol{x}\right\| \left\|\boldsymbol{y}\right\|}$$

Then, we have

We compute the following

$$s\left(d_i, \boldsymbol{c}\right) = \frac{d_i^T \boldsymbol{c}}{\left\|d_i\right\| \left\|\boldsymbol{c}\right\|}$$

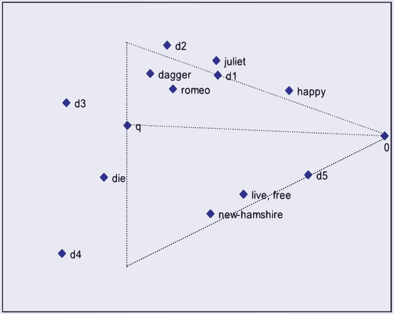
We have the following figure

Then, we have

We compute the following

$$s\left(d_i, \boldsymbol{c}\right) = \frac{d_i^T \boldsymbol{c}}{\left\|d_i\right\| \left\|\boldsymbol{c}\right\|}$$

We have the following figure



We have the following conslusions

Document d_1 is closer to query q than d_5

- As a result d_1 is ranked higher than d_5.

This conforms to our human preference

- Both Romeo and Juliet died by a dagger.

Document d_1 is slightly closer to q than d_2

- d_1 , containing both Romeo and Juliet, is more relevant to the query thand_2

We have the following conslusions

Document d_1 is closer to query q than d_5

- As a result d_1 is ranked higher than d_5.

This conforms to our human preference

- Both Romeo and Juliet died by a dagger.

Document d_1 is slightly closer to q than d_2

- d_1 , containing both Romeo and Juliet, is more relevant to the query thand_2

We have the following conslusions

Document d_1 is closer to query q than d_5

- As a result d_1 is ranked higher than d_5.

This conforms to our human preference

- Both Romeo and Juliet died by a dagger.

Document d_1 is slightly closer to q than d_2

- d_1 , containing both Romeo and Juliet, is more relevant to the query thand_2

Therefore

Latent Semantic Analysis

- It is able to find that d_1 , containing both Romeo and Juliet, is more relevant to the query than d_2